In [1]:

from google.colab import files

uploaded=files.upload()

---------------------------------------------------------------------------

ModuleNotFoundError Traceback (most recent call last)

<ipython-input-1-49432676dcaa> in <module>()

----> 1 from google.colab import files

2 uploaded=files.upload()

ModuleNotFoundError: No module named 'google.colab'

In [0]:

import pandas as pd

import numpy as np

In [0]:

import io

df2=pd.read\_csv(io.BytesIO(uploaded['my\_data.csv']))

In [56]:

print(df2)

400 1075 1066

0 375 1074 1066

1 350 1374 1360

2 340 1300 1300

3 300 1150 1056

4 295 942 956

5 290 960 923

6 270 960 980

7 250 931 970

8 132 895 905

9 100 705 710

10 99 699 704

11 98 699 691

12 95 650 650

13 90 620 634

14 80 620 300

15 72 690 250

16 67 700 223

17 59 739 200

18 45 1078 200

19 30 1278 120

20 270 960 980

21 259 1031 1070

22 132 895 905

23 101 710 720

24 345 1174 1032

25 350 1374 1360

26 340 1300 1300

27 99 649 694

28 98 699 691

29 95 650 650

30 350 1374 1360

31 340 1300 1300

32 310 1250 1156

33 295 942 956

34 351 1374 1360

35 340 1310 1300

36 300 1150 1056

37 295 942 956

38 352 1274 1260

39 340 1300 1300

40 300 1150 1056

41 295 942 956

In [57]:

type(df2)

Out[57]:

pandas.core.frame.DataFrame

In [0]:

x=df2.iloc[:,0]

In [59]:

print(x)

0 375

1 350

2 340

3 300

4 295

5 290

6 270

7 250

8 132

9 100

10 99

11 98

12 95

13 90

14 80

15 72

16 67

17 59

18 45

19 30

20 270

21 259

22 132

23 101

24 345

25 350

26 340

27 99

28 98

29 95

30 350

31 340

32 310

33 295

34 351

35 340

36 300

37 295

38 352

39 340

40 300

41 295

Name: 400, dtype: int64

In [0]:

y=df2.iloc[:,1]

In [0]:

def createA(x, D):

'''

Create the matrix A with degree D

Input:

x: np vector of shape (N,)

D: degree of the model

Output:

A: np matrix of shape (N,D+1)

'''

# YOUR CODE HERE

arr=np.zeros((x.shape[0],D+1))

for i in range(x.shape[0]):

arr[i,0]=1.0

for j in range(1,D+1):

arr[i,j]=float(x[i]\*\*j)

A=arr

return A

In [0]:

def train\_w(A, yd):

'''

Inputs:

A: np matrix of shape (N,D+1)

yd: np vector of shape (N,)

Output:

w: np vector of shape (D+1,)

'''

# YOUR CODE HERE

D=A.shape[1]-1

eta=0.005

w=(2\*np.pi\*np.random.random(D+1,)).reshape(D+1,1)

#print('w',w)

for i in range(A.shape[0]):

x=A[i].reshape(1,A.shape[1])

#print('x',x)

#print('w',w)

y=x.dot(w)

#print('y',y)

yd=np.sin(x[0,1])

err=(y-yd)\*\*2

#print('err',err)

cnt=0

while err>=0.001:

for j in range(w.shape[0]):

w[j]=w[j]-2\*eta\*(y-yd)\*x[0,j]

y=x.dot(w)

err=(y-yd)\*\*2

cnt=cnt+1

#if cnt>200000: break

#print('err',err)

print(err)

print(w)

return w

In [63]:

A=createA(x,3)

from sklearn.preprocessing import StandardScaler

scaler=StandardScaler()

scaler.fit(A)#computes mean and std

A=scaler.transform(A)# x = ( x - mean ) / std

print(A)

[[ 0. 1.30212629 1.59406046 1.89528333]

[ 0. 1.08701732 1.21927858 1.32837793]

[ 0. 1.00097373 1.07660299 1.12304857]

[ 0. 0.65679938 0.54725589 0.41557938]

[ 0. 0.61377759 0.48573997 0.33924211]

[ 0. 0.5707558 0.42525793 0.26544918]

[ 0. 0.39866862 0.19366857 -0.00514168]

[ 0. 0.22658145 -0.02137869 -0.23847312]

[ 0. -0.78873288 -0.95344314 -1.00465074]

[ 0. -1.06407236 -1.1069538 -1.07939776]

[ 0. -1.07267672 -1.11106865 -1.08110554]

[ 0. -1.08128108 -1.11514214 -1.08277917]

[ 0. -1.10709416 -1.12711448 -1.08759857]

[ 0. -1.15011595 -1.14624128 -1.09498002]

[ 0. -1.23615954 -1.18139324 -1.10745733]

[ 0. -1.30499441 -1.20653722 -1.11543545]

[ 0. -1.3480162 -1.22090817 -1.11960327]

[ 0. -1.41685107 -1.24175121 -1.12508777]

[ 0. -1.53731209 -1.27185783 -1.13165728]

[ 0. -1.66637747 -1.29512015 -1.13534441]

[ 0. 0.39866862 0.19366857 -0.00514168]

[ 0. 0.30402068 0.07334549 -0.13790836]

[ 0. -0.78873288 -0.95344314 -1.00465074]

[ 0. -1.055468 -1.1027976 -1.07765548]

[ 0. 1.04399553 1.14742384 1.22422546]

[ 0. 1.08701732 1.21927858 1.32837793]

[ 0. 1.00097373 1.07660299 1.12304857]

[ 0. -1.07267672 -1.11106865 -1.08110554]

[ 0. -1.08128108 -1.11514214 -1.08277917]

[ 0. -1.10709416 -1.12711448 -1.08759857]

[ 0. 1.08701732 1.21927858 1.32837793]

[ 0. 1.00097373 1.07660299 1.12304857]

[ 0. 0.74284297 0.67338938 0.57605943]

[ 0. 0.61377759 0.48573997 0.33924211]

[ 0. 1.09562168 1.23377359 1.34956929]

[ 0. 1.00097373 1.07660299 1.12304857]

[ 0. 0.65679938 0.54725589 0.41557938]

[ 0. 0.61377759 0.48573997 0.33924211]

[ 0. 1.10422604 1.24830996 1.37088174]

[ 0. 1.00097373 1.07660299 1.12304857]

[ 0. 0.65679938 0.54725589 0.41557938]

[ 0. 0.61377759 0.48573997 0.33924211]]

In [64]:

w=train\_w(A,y)

print(w)

[[0.00081134]]

[[ 0.34791401]

[ 0.49726995]

[ 0.89595429]

[-0.5687332 ]]

[[ 0.34791401]

[ 0.49726995]

[ 0.89595429]

[-0.5687332 ]]